REMARKS

The applicants have carefully considered the official action dated December 21, 2006, and the references it cites. Claims 1-41 are pending in this application, of which claims 1, 11, and 27 are independent. In the official action, claims 1-5, 11-22, 26-35, and 37-39 were rejected under 35 U.S.C. § 102(b) as anticipated by Catherwood and claims 1-41 were rejected under 35 U.S.C. § 102(b) as anticipated by Blomgren. By way of this response, the applicants have amended claim 1. No new matter has been added. The applicants respectfully request entry of the amendment to place the application in condition for appeal. In view of the following remarks, the applicants respectfully traverse the rejections and respectfully submit that all pending claims are in condition for allowance. Favorable reconsideration is respectfully requested.

I.   Claims 1-5

The applicants respectfully submit that independent claim 1 is allowable over the art of record. Independent claim 1 is directed to an address generator that includes, *inter alia*, an adder to add a first address component value and a second address component value to generate a first address and a control input that causes the adder to add the first address component value and the second address component value to generate a second address if a correction indicator indicates the first address is incorrect.

The applicants respectfully submit that Catherwood does not describe or suggest a control input to modify an operation of an adder as recited in claim 1. The examiner contends that a control input of a multiplexer (130) taught by Catherwood constitutes the control input recited in claim 1. On the contrary, the applicants respectfully submit that the control input of the Catherwood multiplexer (130) does not modify an operation of the

Catherwood adder (125), but instead controls an operation of the multiplexer (130). There is no evidence in the record to the contrary. Accordingly, Catherwood cannot be construed to describe or suggest a control input to modify an operation of an adder as recited in claim 1. Thus, the applicants respectfully submit that Catherwood does not teach each and every element of claim 1 and, thus, claim 1 cannot be anticipated thereby.

Further, the applicants maintain their position discussed in the previous response. Specifically, as set forth by the antecedent language in claim 1, the first address component value used to generate the first address is the same (i.e., is equal to) the first address component value used to generate the second address, and the second address component value used to generate the first address is the same (i.e., is equal to) the second address component value used to generate the second address. Catherwood does not describe or suggest an adder to add a first address component value and a second address component value to generate a first address and a control input that causes the adder to add the (same) first address component value and the (same) second address component value to generate a second address if a correction indicator indicates the first address is incorrect.

Although Catherwood does describe generating a first address (the no wrap address) and a second address (the wrap address) based on a first address component value (the current address (100) (*see Catherwood*, FIG. 1)) and a second address component value (the offset (105) (*see Id.*)), Catherwood does not teach or suggest adding the current address (100) (a first address component value) and the offset (105) (a second address component value) to generate a no wrap address (a first address) and a control input that causes an adder to add the (same) current address (100) (the first address component value) and the (same) offset (105) (the second address component value) to generate a wrap address (a second address) if a

correction indicator indicates the no wrap address (the first address) is incorrect. Instead, Catherwood teaches using an adder (125) to determine a sum value based on the current address (100) and the offset (105). *See Catherwood*, ¶ 19. Catherwood then uses the sum value to determine the wrap address and the no wrap address and selects the wrap address or the no wrap address as a next address based on a mode bit. *See Id.*, ¶'s 28-31. Specifically, the no wrap address is equal to (or the same as) the sum value generated by the adder (125) which is communicated to an input of a multiplexer (155) (*See Id.*, FIG. 1 and ¶'s 19 and 28-30), and the wrap address is determined based on the sum value using a subtractor (135), logic gates (140 and 145), and a multiplexer (150) that communicates the wrap address to another input of the multiplexer (155) (*See Id.*, FIG. 1 and ¶'s 19 and 28, 29, and 31). The multiplexer (155) selects the no wrap address (which is the sum generated by the adder (125)) or the wrap address (which is generated by the subtractor (135), the logic gates (140 and 145), and the multiplexer (150) based on the sum value).

As discussed above, Catherwood describes generating one sum value one time for each address computation, and describes generating a no wrap address equal to the sum value or a wrap address based on the sum value using additional logic operations. However, Catherwood is completely devoid of any teaching or suggestion that the adder (125) adds the same current address value (100) and the same offset value (105) to determine a second address when a correction indicator indicates the first address is incorrect. Instead, when the Catherwood system determines a subsequent address, the current address value (100) is different from a previously generated current address value (100). This is remarkably different than the language of claim 1, which requires the adder to determine the second address based on the same first address component value and the same second address component value used to generate the first address. In addition, when the Catherwood system

determines the subsequent address, the carry bit values used for the subsequently generated current address are not necessarily the same as the carry bit values used for the previously generated current address. Thus, Catherwood fails to describe or suggest using an adder to add a first address component value and a second address component value to generate a first address and a control input that causes the adder to add the (same) first address component value and the (same) second address component value to generate a second address if a correction indicator indicates the first address is incorrect.

Turning to the rejection of claim 1 over Blomgren, the applicants respectfully submit that Blomgren does not describe or suggest an adder to add a first address component value and a second address component value to generate a first address and a control input that causes the adder to add the first address component value and the second address component value to generate a second address if a correction indicator indicates the first address is incorrect. The applicants respectfully maintain their position that the Blomgren adjustments to generate subsequent addresses do not constitute adding a first address component value and a second address component value to generate a first address and adding the (same) first address component value and the (same) second address component value to generate a second address. Instead, as discussed in the previous response, Blomgren teaches generating a first address and adding adjustment values to the first address to generate subsequent dependent addresses. *See Blomgren*, col. 2, ll. 49-54; col. 4, ll. 55-62; and col. 7, ll. 5-8. Blomgren defines dependent addresses as addresses corresponding to sequentially located instructions that depend on previously generated addresses. *See Id.*, col. 7, ll. 15-18 ("A sequence of more than one PUSH or POP is precisely a sequence of dependent addresses, since the address of one PUSH depends upon the address of the previous PUSH."). Accordingly, for each subsequently generated dependent address, the adjustment values must

be different from one another to access respective operands of the sequentially located instructions. In contrast, claim 1 recites adding a first address component value and a second address component value to generate a first address and adding the (same) first address component value and the (same) second address component value to generate a second address.

The examiner contends that Blomgren describes adding a first address component value and a second address component value to generate a first address and adding the (same) first component value and the (same) second component value to generate a second address by describing adding address values for 16-bit and 32-bit functionality using the same segment_base and then adding index information or displacement information. On the contrary, even if Blomgren describes using the same segment_base value to generate more than one address, the index values change for each generated address and the displacement values change for each generated address. If the Blomgren index values and displacement values were not different for each generated address, the same address would be repeatedly generated causing the same operand value to be retrieved for every instruction. Such an interpretation of Blomgren renders the Blomgren address generation system useless for generating different addresses to retrieve operands for respective instructions. Accordingly, the applicants respectfully submit that Blomgren does not describe or suggest each and every element of claim 1 and, thus, cannot anticipate claim 1.

In view of the above, neither Catherwood nor Blomgren describes or suggests each and every element recited in claim 1. Accordingly, the applicants respectfully submit that independent claim 1 and all claims dependent thereon are in condition for allowance.

II.   <u>Claims 11-26</u>

The applicants respectfully submit that independent claim 11 is allowable over the art of record. Independent claim 11 is directed to an apparatus that includes, *inter alia*, an address generator to generate a first address from a set of address component values and a second address from the set of address component values if the first address is incorrect. Based on the antecedent language used in claim 11, the set of address component values used to generate the first address is the same as (i.e., is equal to) the set of address component values used to generate the second address.

The applicants respectfully submit that Catherwood does not describe or suggest an address generator to generate a first address from a set of address component values and a second address from the (same respective) set of address component values if the first address is incorrect. The examiner contends that Catherwood's circuitry for determining whether an offset is positive or negative or whether address wrapping is required constitutes generating a second address from the same set of address component values used to generate a first address if the first address is incorrect. On the contrary, Catherwood describes unconditionally generating a no wrap address communicated by the adder (125) to an input of a multiplexer (155) and unconditionally generating a wrap address communicated by a multiplexer (150) to another input of the multiplexer (155). The multiplexer (155) then selects one of the no wrap address or the wrap address. Thus, Catherwood does not describe generating a second address (a wrap address) based on the conditional requirement recited in claim 11 that the first address is incorrect. Therefore, Catherwood does not describe or suggest each and every element of claim 11 and, thus, cannot anticipate claim 11.

The applicants respectfully submit that Blomgren does not describe or suggest an address generator to generate a first address from a set of address component values and a second address from the (same respective) set of address component values if the first address is incorrect. The examiner contends that Blomgren describes an address generator to generate a first address from a set of address component values and a second address from the (same respective) set of address component values if the first address is incorrect by describing adding address values for 16-bit and 32-bit functionality using the same segment_base and then adding index information or displacement information. On the contrary, as discussed above in connection with claim 1, such an interpretation of Blomgren renders the Blomgren address generation system useless for generating different addresses to retrieve operands for respective instructions.

Further, as discussed above, the Blomgren adjustment values used to determine each subsequent dependent address must be different from one another so that the dependent addresses can be used to access different operands for respective instructions. In contrast, claim 11 recites an address generator to generate a first address from a set of address component values and a second address from the (same respective) set of address component values. Because the Blomgren adjustment values differ from one another, Blomgren cannot be construed to describe using the same set of address component values to generate a first address and a second address. Accordingly, the applicants respectfully submit that Blomgren does not describe or suggest an address generator to generate a first address from a set of address component values and a second address from the (same respective) set of address component values.

In view of the foregoing, the applicants respectfully submit that neither Catherwood nor Blomgren describes or suggests each and every element recited in claim 11. Accordingly, the applicants respectfully submit that independent claim 11 and all claims dependent thereon are in condition for allowance.

III.   Claims 27-41

The applicants respectfully submit that independent claim 27 is allowable over the art of record. Independent claim 27 is directed to a method that involves, *inter alia*, performing a first addition of a first address component value and a second address component value to generate a first address and modifying an operation in a second addition of the first address component value and the second address component value to generate a second address if the first address is incorrect. Based on the antecedent language used in claim 27, the first address component value used to generate the first address is the same (i.e., is equal to) the first address component value used to generate the second address and the second address component value used to generate the first address is the same (i.e., is equal to) the second address component value used to generate the second address.

The applicants respectfully submit that, at least for the reasons discussed above in connection with claims 1 and 11, neither Catherwood nor Blomgren describes or suggests performing a first addition of a first address component value and a second address component value to generate a first address and modifying an operation in a second addition of the (same) first address component value and the (same) second address component value to generate a second address if the first address is incorrect. Therefore, neither Catherwood nor Blomgren describes or suggests each and every element recited in claim 27.

Accordingly, the applicants respectfully submit that independent claim 27 and all claims

dependent thereon are in condition for allowance.

In view of the foregoing, the applicants respectfully submit that this application is in

condition for allowance. The Commissioner is hereby authorized to charge any deficiency in

the amount enclosed or any additional fees which may be required during the pendency of

this application under 37 CFR 1.16 or 1.17 to Deposit Account No. 50-2455. If there are any

remaining matters that the examiner would like to discuss, the examiner is invited to contact

the undersigned representative at the telephone number set forth below.

Respectfully submitted,

/Mark G. Hanley/

Mark G. Hanley
Registration No. 44,736
Hanley, Flight & Zimmerman, LLC
(at customer number **34431**)
150 South Wacker Drive
Suite 2100
Chicago, Illinois 60606
312.580.1020

Dated: **March 21, 2007**